# REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-02-

0711



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| 28/02/02 | Final Report | 15/01/99 to 11/30/01 |

**4. TITLE AND SUBTITLE**

Collective Mangement of Satellite Clusters

**5a. CONTRACT NUMBER**
F49620-99-C-0009

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Dr. Jorge E. Tierno

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Honeywell Laboratories
3660 Technology Dr.
Minneapolis, MN 55418

**8. PERFORMING ORGANIZATION REPORT NUMBER**

CON-R02-001

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

AFOSR
801 North Randolph Street
Arlington, VA 22203-1977

20020402 074

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.



**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

During this contract we carried out work in three areas relevant to the control of clusters of satellites: orbital analysis, cluster reconfiguration, and cluster control. We developed an approximate separation function, relatively simple to compute, to help predict the short-term behavior of the constellation. We applied optimization methods to develop low-fuel trajectories compatible with operational requirements and market-oriented programming to solve the cluster reconfiguration problem with minimal fuel usage. We also developed a guidance algorithm for the constellation in order to maintain operational formations while minimizing fuel consumption.

**15. SUBJECT TERMS**
Virtual Satellites, Formation Flying, Guidance and Control

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Jorge E. Tierno |
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | | 38 | 19b. TELEPHONE NUMBER *(include area code)* 612 951 7356 |

# Collective Management of Satellite Clusters

Final Report
February 2002

**Abstract**

During this contract we carried out work in three areas relevant to the control of clusters of satellites: orbital analysis, cluster reconfiguration, and cluster control. We developed an approximate separation function, relatively simple to compute, to help predict the short-term behavior of the constellation. We applied optimization methods to develop low-fuel trajectories compatible with operational requirements and market-oriented programming to solve the cluster reconfiguration problem with minimal fuel usage. We also developed a guidance algorithm for the constellation in order to maintain operational formations while minimizing fuel consumption.

**Prepared by:**
Dr. Jorge E. Tierno
Honeywell Laboratories
3660 Technology Dr.
Minneapolis, MN 55418
jorge.tierno@honeywell.com

This document is based upon work supported by AFOSR under contract no. F49620-99-C-0009. Any opinions, findings, conclusions, or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of AFOSR.

# Table of Contents

# List of Figures

# List of Tables

# Section 1 Introduction

## 1.1 Research Summary

This document reports on the work carried out by Honeywell Laboratories from 1999 to 2001 under the AFOSR contract F49620-99-C-0009 entitled "Collective Management of Satellite Clusters."

During this contract we carried out work in three areas: orbital analysis, cluster reconfiguration, and cluster control. These areas correspond to the top two layers of the proposed hierarchy for cluster management shown in Figure 1.1. We did not address problems relevant to the bottom layer, which is concerned with attitude control and regulation. It is important to note, however, that this hierarchical decomposition is to some degree arbitrary and that it leads to a certain loss of performance in exchange for design simplicity and robustness.

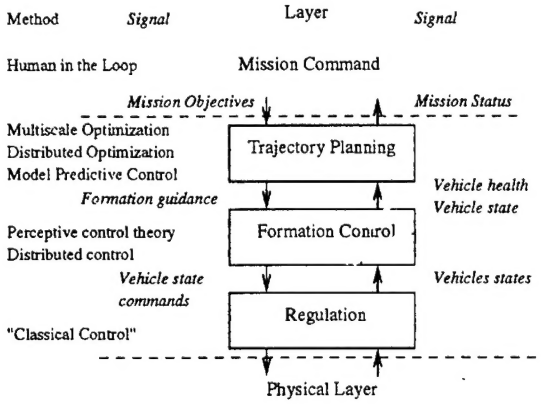


Figure 1.1: Control hierarchy for cluster management

We approached the subject of orbital analysis for cluster maintenance from two points of view. In the work reported in Section 2, we define an approximate separation function, relatively simple to compute, to help predict the short-term behavior of the constellation and to aid in tasks such as collision avoidance. We do further orbital analysis in Section 6, where we show how direct optimization techniques can be used to generate orbital paths that meet mission objective requirements

(cluster shape) and require minimal fuel usage for their maintenance.

In Section 3 we show how market-oriented programming (MOP) methods can be applied to the problem of cluster reconfiguration. Cluster reconfiguration is necessary because maintaining operational formation for long periods of time can be prohitively expensive. It is thus necessary to move the constellation from parking orbit into mission orbit and back with minimal use of fuel. Although suboptimal, MOP is extremely simple to implement and verify, even when the cluster has a changing number of components. In this section we also analyze the robustness of distributed management algorithms to different types of failure modes to guarantee their safety.

Finally, we cover the issues relevant to cluster guidance in Sections 4 and 5. In these sections we present an approach to cluster guidance that achieves very low levels of fuel usage while being easily ammenable to all the operational restrictions of the Techsat21 mission. In section 5 we discuss a detailed study of the effects of different factors, such as sensor noise or constellation shape, on the total fuel requirements of the cluster. For this study, we use the parameters of the Techsat21 demonstrator mission, planned for 2003, to analyze its feasibility and further direct its development.

## 1.2 Personnel

The following scientists have been partially supported by this contract: Dr. Jorge E. Tierno, Dr. Blaise Morton, Dr. John Samson, Nicholas Weininger, Todd Carpenter, Scott Snyder, and Lynn Gravatt.

## 1.3 Publications

"Collective Management of Satellite Clusters", in Proceedings of the 1999 AIAA Conference in Guidance and Control.
"Control of LEO Satellite Clusters", in Proceedings of the 2000 IEEE Conference on Decision and Control.

# Section 2 Orbital Analysis and Satellite Cluster Maintenance

Our approach to orbital analysis is based on instantaneous Kepler orbits. The true paths followed by satellites near the Earth are not Kepler ellipses, owing to forces associated with the non-point-mass nature of the geopotential (e.g., the $J_2$ term), the moon's gravity, atmospheric drag, and other factors. The point-mass geopotential dominates these other effects, usually by several orders of magnitude, so for short periods of time the Kepler-ellipse approximation is a reasonable choice for the nominal motion, about which perturbations may be superimposed. In fact, the Kepler-ellipse approximation may become extremely accurate when used to forecast the *relative* motion of identical spacecraft flying in close proximity (for times on the order of one orbital period), because the perturbative forces tend to affect all the spacecraft in the same way. On the time scale of a single orbital period, the perturbations will cause the Kepler orbits of the various satellites to drift together.

When the time scale is enlarged to, say,100 orbital periods, however, we expect to observe divergences in the parameters of the Kepler ellipses of the various satellites. When the divergences become sufficiently large, it will be necessary to apply corrective thrust to one or more spacecraft to bring the configuration back together. For example, if the energies of the orbits drift apart, there will be no way to keep the flock together because the orbital period is a function of the energy. Thus, a part of the control strategy must be to keep the energies of the orbits close together.

To formulate the complete control strategy the following issues must be addressed:

1. How do we monitor the state of the satellite system?
2. What are the desirable states for the system to be in?
3. How do we determine when a thrust maneuver is necessary?
4. When necessary, how do we select the appropriate maneuver?

One step toward answering these questions is to develop and analyze a separation function for pairs of satellites. The separation function tells how close together and how far apart each pair of satellites is expected to get during the next period (using the Kepler-ellipse approximation). The

values of the separation function for all pairs of satellites will provide one way to monitor the configuration. It may be necessary to consider other functions as well, but certainly we need to monitor bounds on intersatellite distances to keep the system healthy.

To decide when maneuvers are necessary, we will need to determine acceptable ranges for the separation functions. For example, it might be the case that we want to keep the distance between satellites at least 10 m but no more than 1000 m. One question to address in passing is whether a specified range is even achievable. In any case, the basic idea is to mandate a maneuver whenever one pair of satellites is outside the acceptable range for the separation function.

## 2.1 Notation and Coordinates

At each instant of time, a satellite's position and velocity vectors are associated with the parameters of a Kepler orbit about the Earth's center of mass consistent with Newton's law. We will be working with orbiting satellites for which the position vector $\mathbf{x}$ and the velocity vector $\mathbf{v} = \frac{d\mathbf{x}}{dt}$ are linearly independent, so the angular momentum $\mathbf{h}$ (per kilogram) defined by $\mathbf{h} = \mathbf{x} \times \mathbf{v}$ is a nonzero vector. If the point-mass geopotential were the only force acting on the satellites, then each satellite's $\mathbf{h}$ would be time invariant. The instantaneous Kepler motion of each satellite is confined to the plane through the linear subspace normal to its $\mathbf{h}$ vector.

### 2.1.1 Velocity Circle

Pick a coordinate system $\mathbf{x} = (x, y, z)$ such that $\mathbf{h}$ points along the positive z-axis. Then from $\mathbf{x(t)} \times \mathbf{v(t)} = \mathbf{h} = (0, 0, h)$ we see that

$$\mathbf{x}(t) = (r(t)\cos(\theta(t)), r(t)\sin(\theta(t)), 0)$$

Differentiating this last equation with respect to t and taking the cross product with $\mathbf{x}(t)$, we find

$$(0, 0, h) = \mathbf{h} = \mathbf{x} \times \frac{d\mathbf{x}}{dt} = (0, 0, r(t)^2 \frac{d\theta}{dt})$$

So in particular,

$$\frac{d\theta}{dt} = h/r(t)^2$$

Now, from Newton's equation $\frac{d^2\mathbf{x}}{dt^2} = -k\frac{\mathbf{x}}{||\mathbf{x}||^3}$ where $k = GM_e$ and setting $R = k/h$, we obtain after integration:

$$\mathbf{v} = R(-\sin(\theta), \cos(\theta), 0) + \mathbf{c}$$

where $\mathbf{c} = (c_1, c_2, 0)$ is an integration constant. Interpreting this last equation, we see that the velocity vector $\mathbf{v}$ moves along a circle centered at $\mathbf{c}$, with radius $R = k/h$, and lies in the plane through the origin which is orthogonal to $\mathbf{h}$. Also, note that $\mathbf{v} - \mathbf{c}$ is always orthogonal to the position vector $\mathbf{x} = r(\cos(\theta), \sin(\theta), 0)$. (See [3] for more details.) Specifically,

$$\mathbf{x} = r\frac{\mathbf{v} - \mathbf{c}}{R} \times \frac{\mathbf{h}}{h}$$

### 2.1.2 Eccentricity, True Anomaly, and Energy

Defining $\epsilon = ||\mathbf{c}||/R$, choose coordinates $(u, v)$ for the plane of the velocity circle so that the center of the circle lies on the positive $v$-axis. Then $\epsilon$ is called the eccentricity and

$$\mathbf{v} = R(-\sin(\theta), \epsilon + \cos(\theta), 0)$$

Substituting into $\mathbf{h} = \mathbf{x} \times \mathbf{v}$, we obtain

$$h = rR(1 + \epsilon\cos(\theta))$$

and

$$r = \Lambda/(1 + \epsilon\cos(\theta)) where \quad \Lambda = h^2/k$$

The parameter $\theta$ defined as above is commonly called the true anomaly of the satellite. Perigee ($r = r_{min}$) occurs when $\theta = 0$.

Another classic invariant is the energy $E$ (per kilogram) defined by

$$E = \frac{1}{2}\mathbf{v} \cdot \mathbf{v} - k/r$$

An easy computation shows that $\frac{dE}{dt} = 0$ and

$$2E = ||\mathbf{c}||^2 - R^2 = -(1 - \epsilon^2)k^2/h^2 \tag{2.1}$$

## 2.2 Characterization of Orbits with Fixed Energy and Eccentricity

Our nominal strategy is to work with orbits of fixed energy and eccentricity. Here we describe these orbits in terms of velocity-circle coordinates and examine impulsive orbit transfers that preserve these two parameters.

If we fix both $E$ and $\epsilon$ of the orbits, from the definition of $\epsilon = ||\mathbf{c}||/R$, we find

$$2E = R^2(\epsilon^2 - 1)$$

Thus,

$$R = \sqrt{2E/(\epsilon^2 - 1)} \quad \text{and} \quad ||\mathbf{c}|| = \epsilon R$$

show that both $||\mathbf{c}||$ and $R$ are uniquely determined. Also note that the magnitude of the angular momentum $h = k/R$ is determined.

These equations clearly identify geometrically the set of orbits of fixed energy and eccentricity. In the velocity space, the four-parameter set of orbits is obtained as follows:

1. Start with the velocity circle of any one such orbit.
2. Apply an orthogonal transformation (three parameters).
3. Rotate the circle about its center (one parameter).

The orthogonal transformations alter the orbital-plane parameters (classically, these are the inclination, longitude of the ascending node, and argument of perigee) and the direction of the satellite along the orbit. The rotation about the center of the velocity circle is a shift of the true anomaly $\theta$.

We can now characterize those impulsive transformations that preserve both energy and eccentricity. First, because the transfer is impulsive, the position vector $\mathbf{x}$ remains fixed (only the velocity $\mathbf{v}$ changes). But now:

1. The energy is fixed, so $||\mathbf{v}||$ is constant.
2. Because $h$ is constant, $||\mathbf{x} \times \mathbf{v}||$ is constant.

The set of $\mathbf{v}$ restrictions satisfying 1 and 2 lies in two circles in planes orthogonal to $\mathbf{x}$. The only transformations of the velocity vector satisfying 1 and 2 simultaneously are composed of the following two transformations:

1. Rotate the velocity vector by an arbitrary angle $\delta$ about the $\mathbf{x}$ vector (the direction in the plane of the velocity circle through 0 and orthogonal to $\mathbf{v} - \mathbf{c}$).
2. Reflect $\mathbf{v}$ with respect to the plane perpendicular to $\mathbf{x}$.

Transformations of the first type are reasonable and easy to compute mathematically. They can be performed practically for small values of $\delta$. Transformations of the second type may be feasible if the angle between $\mathbf{v}$ and $\mathbf{x}$ is close enough to 90 degrees.

In summary, we can explicitly parameterize the velocity circles corresponding to orbits of fixed energy and eccentricity as the product space of the three-dimensional orthogonal group and the circle. Furthermore, it is easy to compute (using rotations and reflections) all single-impulse $\Delta\mathbf{V}$ commands that preserve the same two parameters.

### 2.2.1 More General Orbit Transfers

To parameterize the more general set of single-impulse orbit transfers, in which it is desired to correct the total energy and eccentricity as well as adjust the range of a separation function, start by finding a single $\mathbf{v_{new}}$ having the desired E and $\epsilon$ parameters for the current $\mathbf{x}$. To do this, we need to solve two quadratic equations in three unknowns. Geometrically, $\mathbf{v_{new}}$ lies in the intersection of a cone and a sphere, easily found by analytic methods:

$$||\mathbf{v_{new}}||^2 = 2(E + k/||\mathbf{x}||)$$

$$||\mathbf{x} \times \mathbf{v_{new}}||^2 = {h_{new}}^2 = \sqrt{\frac{k^2(\epsilon^2 - 1)}{2E}}$$

To obtain a more general set of transformations,take any solution $\mathbf{v_{new}}$ to these equations and apply to it the general set of transformations of $\mathbf{v_{new}}$, preserving E and $\epsilon$ as discussed in the previous subsection, to obtain the more general set of transformations.

## 2.3 Separation Function

A separation function is a map from pairs of satellites to $\mathbf{R^2}$, the real plane. The first coordinate is an estimate of how close the two satellites will come during the next orbital period, the second coordinate is an estimate of how far apart they will drift.

A good separation function has three properties:

1. The first value vanishes whenever the two satellites are on a collision course.
2. The second value is approximately equal to the maximum separation distance during the next orbital period.
3. It is analytically tractable.

The significance of the first two properties should be clear. Indeed, the most obvious selection for the separation function (called the obvious function) is constructed as follows: simulate the Kepler orbits of the two satellites over the next orbital period and compute the maximum and minimum distance separations during that period. Unfortunately, the obvious function is less tractable than we want, thus violating the third property. From a theoretical perspective, the difficulty with the obvious function is the transcendental nature of the true anomaly as a function of time (except when $\epsilon = 0$, discussed below).

As a partial alternative to the obvious function, we are experimenting with a candidate first coordinate for a separation function called the cross-path delay. The cross-path delay is defined as

follows: given the two orbits, compute the line on which the two orbital planes intersect. There are two pairs of points, $(\mathbf{x_{1a}}, \mathbf{x_{2a}})$ and $(\mathbf{x_{1b}}, \mathbf{x_{2b}})$ (subscript 1 corresponds to the first orbit, subscript 2 to the second), near to each other on this line of intersection. Note that if the centers of gravity of the two satellites are going to collide anywhere, it would have to be at one of these two pairs of points. The cross-path delay of the first pair of points is the difference in the times when satellite 1 will be at $\mathbf{x_{1a}}$ and when satellite 2 will be at $\mathbf{x_{2a}}$. The cross-path delay of the second pair is defined similarly, and the overall cross-path delay is the minimum of these two values.

There are two attractive features of the cross-path delay function. First, it should be a good measure of the *apparent* separation (as seen from the Earth) between the two satellites when they approach the crossing nodes. Second, the cross-path delay is a readily computable function. The approach to the computation is as follows: note the current true anomalies, $\theta_1 and \theta_2$, of the two satellites and compute the true anomalies corresponding to the crossing points (two on each orbit). It is possible to derive an analytic expression for time as a function of the true anomalies. Given

$$\frac{d\theta}{dt} = \frac{h(1+\epsilon\cos(\theta))^2}{\Lambda^2}$$

we can rewrite this as

$$\int dt = \frac{\Lambda^2}{h}\int \frac{d\theta}{(1+\epsilon\cos(\theta))^2} \tag{2.2}$$

and the right side of this equation (2.2) can be evaluated:

$$\int \frac{d\theta}{(1+\epsilon\cos(\theta))^2} = \frac{\epsilon\sin(\theta)}{(\epsilon^2-1)(1+\epsilon\cos(\theta))} + \frac{2}{(1-\epsilon^2)^{\frac{3}{2}}}\arctan\frac{\sqrt{1-\epsilon^2}\tan(\frac{\theta}{2})}{1+\epsilon}$$

The cross-path delay can be computed accurately from this expression. For orbits having eccentricities around 0.0001 or smaller, there are approximate solutions to the above integral that are far easier to evaluate. The approximate solution to first order in $\epsilon$ is

$$\int dt = \frac{\Lambda^2}{h}(\theta - 2\epsilon\sin(\theta))$$

while the solution to second order in $\epsilon$ is

$$\int dt = \frac{\Lambda^2}{h}(\theta - 2\epsilon\sin(\theta) + 3\epsilon^2(\frac{\theta}{2} + \frac{\sin(2\theta)}{4}))$$

# Section 3 Satellite Cluster Reconfiguration

In the course of a mission such as that outlined in the TechSat21 program, a satellite cluster may have to undergo repeated reconfiguration.

Repeated reconfiguration of satellites presents the problem of equalization of fuel use across the cluster. Naive reconfiguration algorithms (e.g., each satellite has a fixed station that it takes at reconfiguration time) may produce very unequal fuel use, resulting in a suboptimal total cluster lifetime. We wish to equalize fuel use across a sequence of many maneuvers and do it in a distributed fashion requiring minimal computational resources. We assume that for $n$ satellites in a cluster, there are $n$ known orbital stations to be occupied in a reconfiguration, and that each satellite can calculate its cost to move to each of the stations. Our problem thus becomes assigning satellites to stations so as to maximize cluster lifetime.

## 3.1 Market-Oriented Programming

Wellman [6] has developed an auctioneering model based on the economic theory of general equilibrium. His model has been applied to problems of transportation, computing resource allocation, and product design (see for example [5]).

We investigated the use of *market-oriented programming* techniques to select the stations that satellites occupy during reconfigurations. In our study problem, we have a set of $n$ homogeneous agents, the satellites, each of which must "produce" one of $n$ goods, i.e., stations. There is just one resource, fuel, and each satellite has a fixed, remaining allocation of that resource; they cannot trade fuel among themselves. For this reason, we decided to begin with a highly simplified auction algorithm. In this algorithm, each satellite simply submits a fixed bid for each station. This bid is based on a cost function of the following form $B_{i,j} = \frac{f(\Delta v_{i,j})}{g(rf_i)}$, where $B_{i,j}$ is the bid submitted by satellite $i$ for station $j$, $\Delta v_{i,j}$ is the $\Delta v$ required for satellite $i$ to reach station $j$, and $rf_i$ is the remaining fuel for satellite $i$.

All satellites bids are then submitted to an auction (which might be run on one of the satellites or at a ground station). This auction then assigns satellites to stations using the following algorithm:

1. Determine the lowest bidder for all stations yet to be assigned.

2. Determine the maximum among stations of the lowest bids.
3. Assign the station corresponding to this maximum to its lowest bidder.
4. Remove the assigned station and satellite from the auction.
5. If not all stations have been assigned, go to step 1.

## 3.2 Choosing a cost function

As discussed above, the cost function used to determine bids must increase as the $\Delta v$ required increases and must increase as the fuel remaining decreases. Therefore, the simplest possible function is $B_{i,j} = \frac{\Delta v_{i,j}}{rf_i}$; i.e., the bid is simply the cost as a fraction of fuel remaining. We could also introduce an exponential dependence on fuel remaining, for example, $B_{i,j} = \frac{\Delta v_{i,j}}{\exp(rf_i)}$ or an exponential dependence on $\Delta v$: $B_{i,j} = \frac{\exp(\Delta v_{i,j})}{rf_i}$. Of course, there are numerous other variations possible. We tested the above three functions to determine whether using an exponential dependence on one of the two main parameters increased the effectiveness of fuel equalization.

## 3.3 Test problem

As an initial test of our auction algorithm, we used the following reconfiguration maneuver: move from a parking orbit (i.e., all satellites in the same circular orbit, Figure 3.1) to the mission configuration (Figure 3.2). The parking orbit consisted of 17 satellites spaced out in a circular, polar orbit with an altitude of 637 km and an intersatellite separation of 44 m. (To gauge the effects of different perturbations we added one satellite at the constellation's center).

For a single satellite, then, a transfer from the parking orbit to one of the elliptical orbits in the "circle" required an in-plane burn to change orbital eccentricity and an out-of-plane burn to shift the orbital plane very slightly. The estimated costs for these maneuvers ranged from 4 cm/sec to 83 cm/sec. Each satellite was initially assumed to have sufficient fuel for a $\Delta v$ of 100 m/s; this is consistent with a fuel mass fraction of 10% and a specific impulse of 100 s.

The results show that the simple auction algorithm, using the simplest cost function $\Delta v_{i,j}/rf_i$, produced a near-optimal assignment of configurations among the satellites. The satellites were able to perform 242 reconfigurations before one of them ran out of fuel, and after those reconfigurations only 0.6% of the total fuel was left unused. By comparison, a naive algorithm that sets bids equal to $\Delta v$ regardless of fuel remaining, and thus always produces the same assignment of satellites to stations, resulted in a satellite running out of fuel after 141 reconfigurations, with 42.3% of the

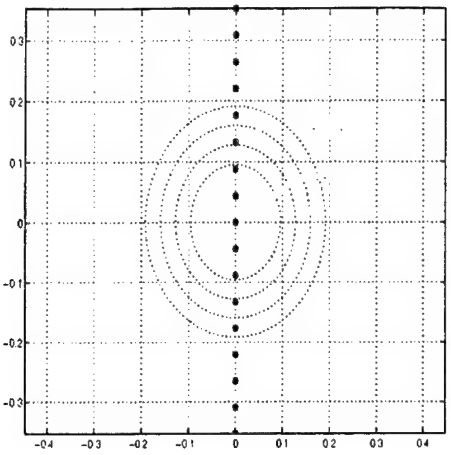


Figure 3.1: Formation before reconfiguration (distances in km)

total fuel unused. The fuel usage histories for the satellites in these two tests are plotted in Figures 3.3 and 3.4.

The cost function $\exp(\Delta v_{i,j})/rf_i$ still ran out of fuel after 242 reconfigurations, but achieved a slightly more uniform fuel distribution, with only 0.4% of the fuel left unused. The function $\Delta v_{i,j}/\exp(rf_i)$, however, did very poorly, running out of fuel after 142 reconfigurations with 41.8% of the fuel left unused. A slight variation on this function, setting the bid to $\Delta v_{i,j} \times \exp(1/rf_i)$, achieved much better results. It ran out of fuel after 239 configurations with only 1.9% of the fuel still unused. The fuel usage histories for these tests are plotted in Figures 3.5, 3.6, and 3.7. All of these results are summarized in Table 3.1.

Table 3.1: Summary of different bidding strategies

| Cost Function | # of Reconfig. | % of Fuel remaining |
|---|---|---|
| $\Delta v_{i,j}/rf_i$ | 242 | 0.6 |
| $\Delta v$ | 141 | 42.3 |
| $\exp(\Delta v_{i,j})/rf_i$ | 242 | 0.4 |
| $\Delta v_{i,j}/\exp(rf_i)$ | 142 | 41.8 |
| $\Delta v_{i,j} \times \exp(1/rf_i)$ | 239 | 1.9 |

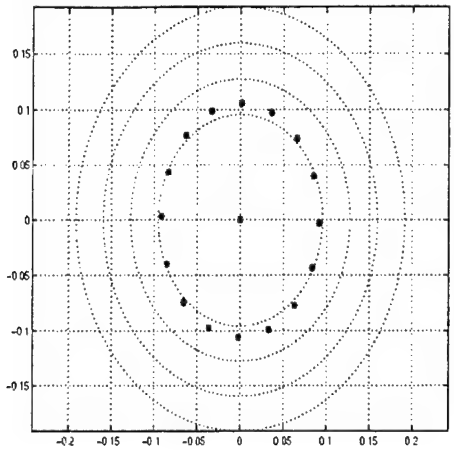


Figure 3.2: Formation after reconfiguration (distances in km)

## 3.4 Distributed Auctioneering for Fault Tolerance

The 1999 TechSat21 program developed an auctioneering algorithm to optimize the station keeping and constellation reconfiguration decisions and to maximize the lifetime of the constellation. The 2000 auctioneering work focused on the distribution of that algorithm to provide fault tolerance and load balancing, while minimizing communication and computation overhead.

This study covered several distinct distribution mechanisms. The best mechanism depends on the number of satellites, the level and type of fault tolerance desired, as well as the available communication and computing bandwidth. For instance, a very small constellation of 2 to 3 satellites would probably use a simple distribution scheme with limited fault coverage. A medium sized constellation of 4 to 36 satellites might use a scheme that is Byzantine resilient. Extremely large constellations would probably apply a hierarchy of approaches, due to the n3 overhead of the Byzantine approach.

The key assumptions in this effort are as follows:

1. Relative satellite positions are available (GPS)
2. Sufficiently precise, synchronized time is available (GPS).
3. Significant on-board processing is available.
4. Satellites have homogeneous capabilities.
5. A reliable communications infrastructure is in place (e.g., a garbled message is detectable).

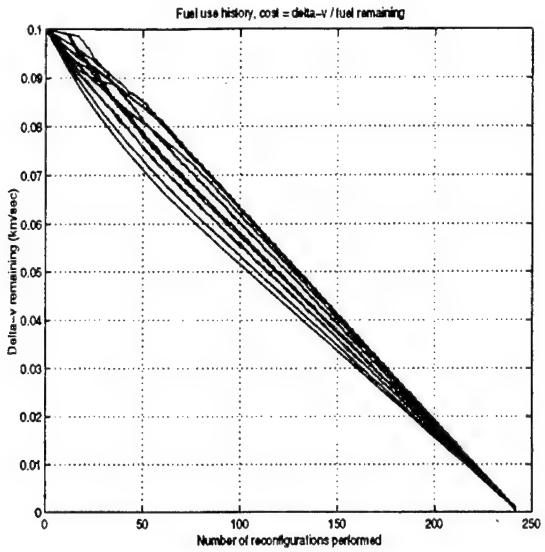


Figure 3.3: Fuel usage history with cost function $\Delta v_{i,j}/rf_i$

A distribution algorithm is considered "safe" if, for a given type of failure, two satellites cannot attempt to assume the same station in the constellation. The failure modes that were considered include, but are not limited to the following: passive satellite failure, communication problems (detectable by the communication subsystem), loss of GPS (time or position errors), sensor failures (i.e., errors in position or fuel measurement), byzantine (asymmetric) failures. For example, if an errant satellite A sent different (but self-consistent) status information to satellites B and C, then B and C could come to conflicting solutions. It has been proven that systems with less than four contributors and two exchanges cannot be made resilient to this class of fault (see [2]).

The following are the reconfiguration approaches analyzed under this program:

Master/Slave: In this approach, developed under the 1999 TechSat21 program, a master collects bids from everyone and distributes the results. This is not resilient to arbitrary failures, passive or otherwise.

Distributed Minimized Computation: This approach distributes the proposed master/slave algorithm, which handles arbitrary fail passive behavior. It attempts to minimize overall computation cycles by having each satellite receive bids from all other satellites, do the bid selection itself, and then act on the result.

Distributed Minimized Communication: This approach distributes the proposed master/slave algorithm, which handles arbitrary fail passive behavior. It attempts to minimize communication overhead by having each satellite broadcast its state. Each satellite then receives these states,

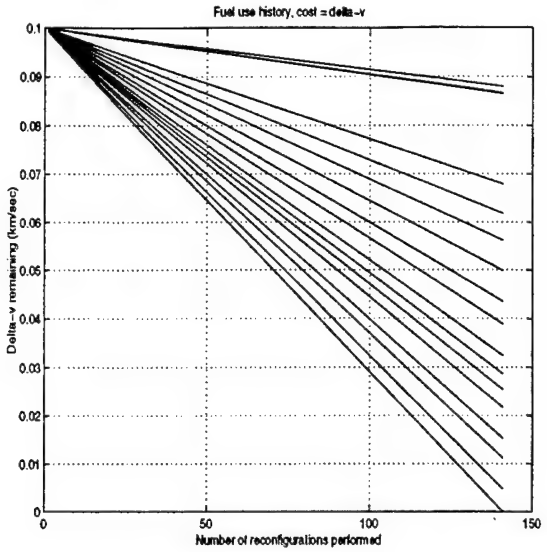


Figure 3.4: Fuel usage history with cost function $\Delta v$

computes bids for itself and the other satellites, does the bid selection, and then acts on the result.

Distributed Dependable: This approach distributes the resulting assignment vector from a distributed algorithm, includes total cost with the assignment vector, and selects the vector that is repeated most often. It then breaks ties by selecting the lowest cost solution and then breaks ties with that by selecting the one provided by the lowest identification number. This one is Byzantine-fault secure for formations of four and greater. Byzantine resilience needs k+1 rounds of information exchange to tolerate k faults. This approach has two rounds and is therefore tolerant to just one Byzantine fault.

Distributed Byzantine Resilient: This approach (which can also optimize for plume avoidance) uses the distributed dependable algorithm, but distributes the result after each assignment and collects new bids following each assignment. This enables enhancements for handling additional Byzantine faults and also allows path calculations for collision and plume avoidance to be factored into cost. At each step, this approach only involves the satellites that do not yet have station assignment, which provides overall lower communication costs.

Another approach is to combine instances of the above, for example, distributed or master/slave at one level and Byzantine resilient at another. This is potentially attractive since the Byzantine-resilient algorithms have significant nonlinear overheads so; large clusters might be unachievable without a hybrid. For instance, the Byzantine solution for a 256-element formation is 126 times slower than the proposed minimally communication-intensive algorithm, and it requires 1840 times

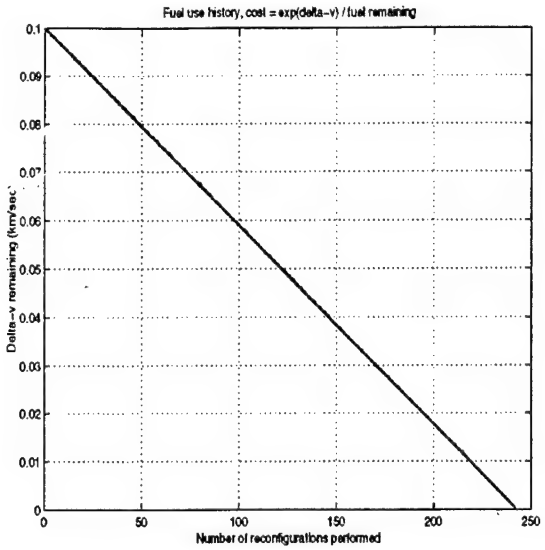


Figure 3.5: Fuel usage history with cost function $\exp(\Delta v_{i,j})/rf_i$

the communication bandwidth.

However, these hierarchical approaches suffer from additional design complexity and increased cost for smaller constellations over the more straightforward approach. Issues include domain selection and arbitration. Examples of such hierarchical approaches include:

1. Holding an election for a master.
2. Holding an election for several masters.
3. Voting results with Byzantine exchange.
4. Applying a Byzantine approach to a cluster of satellites, then applying a Byzantine approach to the clusters (this has unresolved issues with cluster selection).
5. Bidding for a region (stations $\ll$ satellites), with multiple winners per region making up a cluster, then bidding for spots within that cluster.
6. Clustering based on location in the current configuration (some locality benefits, but has potential boundary effects.)

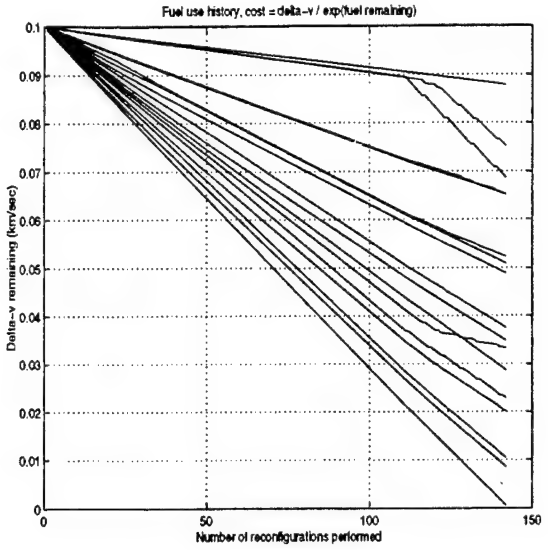


Figure 3.6: Fuel usage history with cost function $\Delta v_{i,j} / \exp(rf_i)$

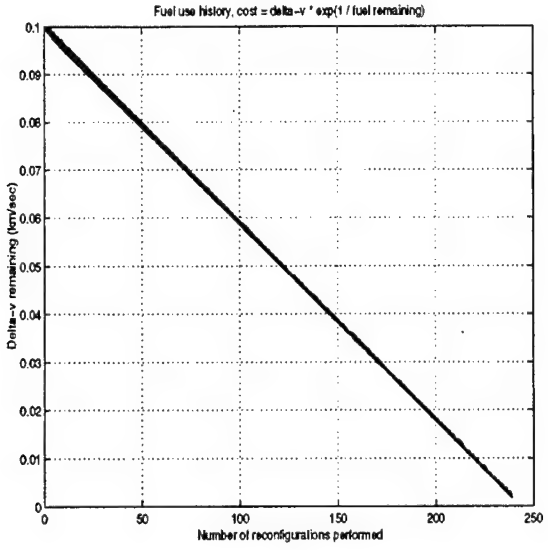


Figure 3.7: Fuel usage history with cost function $\Delta v_{i,j} \times \exp(1/rf_i)$

# Section 4 Formation Control

## 4.1 Perceptive Control Theory

Our formation control layer algorithms are based on perceptive control theory [1]. This theory has been successfully applied to formation control of multiple autonomous agents. In perceptive control theory, the key component of a system, such as a formation of multiple satellites, is defined to be the *perceptive action reference*. This action reference is computed on-line based on sensory measurements. The on-line planner of the system generates the desired state values for the system according to the computed action reference. In addition, the action reference is calculated near or at the same rate as the feedback control. In other words, the action plan is adjusted at a high rate, which enables the planner to handle unexpected or uncertain discrete/continuous events or planned system reconfigurations. Furthermore, unlike traditional methods that require controller replanning to complete the task, once the event or reconfiguration is over, the perceptive control in this case does not require any replanning.

A formation of multiple spacecrafts has a special action reference. The tasks are synchronized according to the given action reference. A perceptive frame will be a mathematical abstraction of these action references. It is a projection from the position $p \in \mathcal{P}$ and the force $f \in \mathcal{F}$ to a reference $s \in \mathcal{S}$, $\Pi : \mathcal{P} \times \mathcal{F} \mapsto \mathcal{S}$. Based on this projection, an action plan of a multi-vehicle formation in the perceptive frame is parameterized by the corresponding action reference parameter. Since the action reference is a function of the real-time sensory information, the desired quantities generated by the action plan are also directly related to the measured data. This creates a mechanism to provide information in the form of numerical values for the base plan using the measurements, and the information is interpreted through a perceptive frame to determine the control input value. Thus, the planning becomes a closed-loop, real-time process. The plan (desired action) can be perceived as an "abstract" entity, like a function. It only has a "real" value when measurements are made within the given perceptive frame.

## 4.2 Application of Perceptive Control Theory to Cluster Formation Flying

Perceptive control theory is a general method for controller design of formations of multiple autonomous agents. A simplified schematic of the control architecture is shown in Figure 4.1.

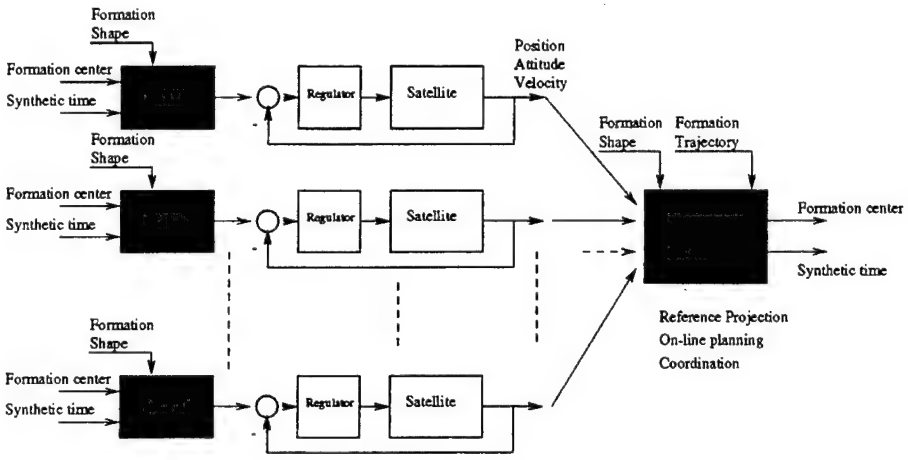


Figure 4.1: Formation control layer architecture

To develop a formation control scheme for multiple spacecraft vehicles, the perceptive control method will be generalized to a broader sensing-based perceptive context. The first step is to find an appropriate perceptive action reference that can efficiently represent and transfer the output measurements of each vehicle to all action planners. The second step is to develop a computational scheme for determining the perceptive reference based on real-time sensor measurements. The third step is the design of a general-purpose controller with respect to the perceptive reference frame.

In our approach, we will compute a perceptive reference variable—synthetic time—that is communicated to the individual vehicles, which will use this variable to generate local guidance commands. Given the current state of the set of vehicles $s_1(t), s_2(t), \ldots, s_n(t)$ and their preplanned trajectory indexed in a parameter $\tau$, $\sigma_1(\tau), \sigma_2(\tau), \ldots, \sigma_n(\tau)$, the reference variable $\theta$ is computed as $\theta = \arg\min_\tau(\|(s_1(t), s_2(t), \ldots, s_n(t)) - (\sigma_1(\tau), \sigma_2(\tau), \ldots, \sigma_n(\tau))\|_P)$. It can be shown that for norms $P$ that meet a certain set of conditions, the system represented in Figure 4.1 is stable in the sense of Lyapunov if the individual components are stable. Once $\theta$ is determined, each individual satellite regulates to the command $\sigma_i(\theta)$. The nominal trajectories can be stored in each vehicle relative to a cluster center. In this way, the trajectory is separated into two components, one describing the overall behavior of the cluster and one describing its shape.

## 4.3 Virtual Leader Reference Projection

For our example, we use the position of the "center" of the constellation as the reference variable. Since the trajectories we are considering are symmetric, we computed the center by taking the average of the current positions. Let $P_i$ be the absolute position of the $i^{th}$ satellite at time $t$, and let $P_{ni}(\theta)$ be a parameterization of the nominal trajectory of the $i^{th}$ satellite. Then, the measured center at time $t$ is given by $C(t) = \frac{1}{n}\sum_1^n P_i$, and a parameterization of the nominal trajectory for the constellation center is $C_n(\theta) = \frac{1}{n}\sum_1^n P_{ni}(\theta)$. The reference projection is then obtained by finding the point in the nominal trajectory for the constellation center closest to the current center. We used as distance the geometric distance (i.e., only the position states were used in the projection). Figure 4.2 shows how this is done for the case of a constellation of two satellites. The reference variable will be the time index $\theta$ that minimizes the distance between the nominal center at time $\theta$ and the current position of the center. We will denote this projection by $\Pi$. $\theta = \Pi(C) = \arg\min\left(\|C(t) - C_n(\theta)\|\right)$. This approach has the advantage of not requiring the setup of a master/slave architecture. We are thus able to have a completely symmetric formation control layer. The benefits of this are simpler architecture and more fault tolerance.

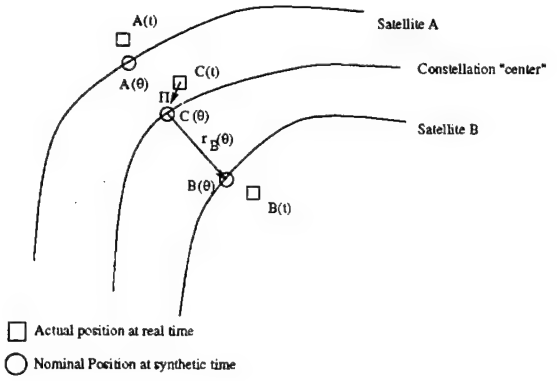


Figure 4.2: Reference projection using constellation center

## 4.4 Use of Relative Position Sensing

Relative position between satellites can be obtained with much higher accuracy than absolute position. Since we are tracking a nominal trajectory given in absolute position for each satellite, it may seem like we will suffer high errors. However, by expressing the nominal trajectories for each satellite relative to the constellation center, and by computing the constellation center using both

relative and absolute position, we can easily work around this problem. In this case, each satellite will compute a reference projection. As we shall see, this will not significantly affect the shape of the formation. If $P_i$ is the absolute position of the $i^{th}$ satellite, the relative position between the $i^{th}$ and $j^{th}$ satellites will be given by $P_{ij} = P_j - P_i$. Let $\epsilon_i$ be the error in absolute position for the $i^{th}$ satellite. Then the satellite can compute the measured center of the constellation as:

$$C_{mi} = \frac{1}{n}\left(n\left(P_i + \epsilon_i\right) + \sum_{i=2}^{n} P_{ij}\right)$$

The satellite now computes its position command $P_i^c$ by summing the desired relative positions from nominal center to the measured position. Knowing past inputs and outputs allows us to estimate current state and model parameters position of the center $P_i^c = r_i\left(\Pi\left(C + \epsilon_i\right)\right) + C + \epsilon_i$. We can approximate this expression by linearizing the projection, $P_i^c \approx r_i\left(\Pi\left(C\right)\right) + \nabla r_i\left(\Pi\left(C\right)\right)\epsilon_i + C + \epsilon_i$. The measured tracking error $\bar{e}$ will now be $\bar{e} = P_i^c - P_{im} \approx e + C(t) - C\left(\Pi(C)\right) + \nabla r_i\left(\Pi\left(C\right)\right)\epsilon_i$, where $e$ is the actual tracking error. Forcing the measured tracking error to be zero, $\bar{e} = 0 \Rightarrow e = -\left(C(t) - C\left(\Pi(C)\right) + \nabla r_i\left(\Pi\left(C\right)\right)\epsilon_i\right)$. The real tracking error thus has two components: one that doesn't depend on the satellite and thus doesn't affect the constellation shape, and one that depends on the measurement error and the gradient of the nominal trajectory. Since this will be a small number, it will dampen the effect of the measurement errors.

# Section 5 Tradeoff Studies

In this section we explore the effect of different factors on the overall fuel usage of the constellation while maintaining formation. To prepare for the demonstration flight of the Techsat21 mission in 2003, it is necessary to establish which are the most fuel-efficient strategies to operate the cluster while in formation. It is also important to determine under which strategy the demonstrator mission will last up to a year. Of that length of time, only a small fraction will be used to test formation flying capabilities. The total fuel budget for each satellite in the demonstrator mission is equivalent to a total $\Delta V$ of 65m/s.

A key metric for the control strategy is thus the constellation endurance, measured by the length of time that the cluster can be maintained in a given formation. All test results are for a reference orbit having a 400 km altitude, 30 deg inclination, and a 2 mm/s relative velocity error, with four satellites per cluster. We tested two formations centered around this reference orbit: an out-of-plane formation, where the projection of the satellites on a horizontal plane centered at the reference orbit lie in a circle of 1km diameter (Figure 5.1), and an in-plane formation, where all satellites lie in an ellipse centered on the reference orbit and with semi-major axes of 1 km along track and 2 km in the radial direction. These are challenging formations, likely to be more stringent than what would be needed for mission purposes.

## 5.1 Effects of Correction Strategy

An important factor on the total fuel usage of the constellation is the rate at which corrections to the orbits are planned and executed. We analyzed the effects of two parameters: the time between orbit corrections and the time allowed for the correction. In the following figures, we present the results obtained with two of the strategies analyzed. One is a correction every two orbits, allowing for a whole orbit to finish the correction. The other is for a correction every orbit, allowing for a half orbit to finish the correction. Figures 5.4 and 5.7 show the fuel usage for each of the four satellites. Figures 5.2, 5.3, 5.5, and 5.6 show the relative distance between two of the satellites in the constellation. To meet the performance requirements of the Techsat21 mission, this distance has to deviate less than 10% from the nominal. The nominal distance for Figures 5.2 and 5.5 is 1.4 km. For Figures 5.3 and 5.6 it is 2 km. As a reference we can compare these values with

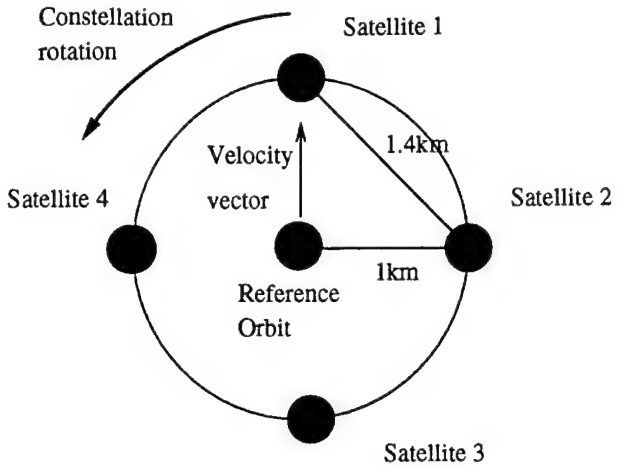


Figure 5.1: Nominal location of satellites and relative distances

those corresponding to an in-plane formation, which minimizes the effects of J2 and higher order perturbations. These results are shown in Figures 5.8 and 5.9.

## 5.2 Effects of Restricting Burns to a Short Interval

The total fuel usage of the constellation will also be influenced by operational constraints that only allow usage of the engines at specific times. For thermal reasons, the engines can only be used for a limited duration (5 to 10 min) and then require up to a five hour cooldown.

This results in a loss of flexibility for the linear program, which leads to lower system performance. To gauge the impact of this effect, we ran two test cases. For these tests, we used the control strategy that gave the best performance tradeoff in the unrestricted-engine-use case.

We ran these tests for the two formations discussed earlier. The results for the more challenging out-of-plane formation are shown in Figures 5.10, 5.11, and 5.12. These are to be compared to Figures 5.2, 5.3, and 5.4, respectively. The results for the in-plane formation are shown in Figures 5.13 and 5.14.

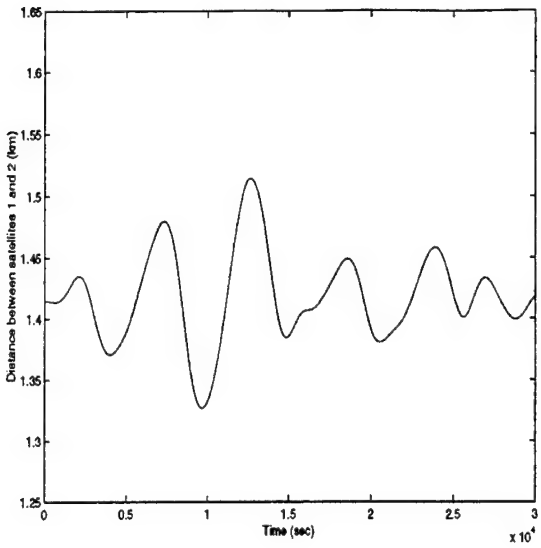


Figure 5.2: Distance between satellites 1 and 2: unrestricted burns, correction every other orbit

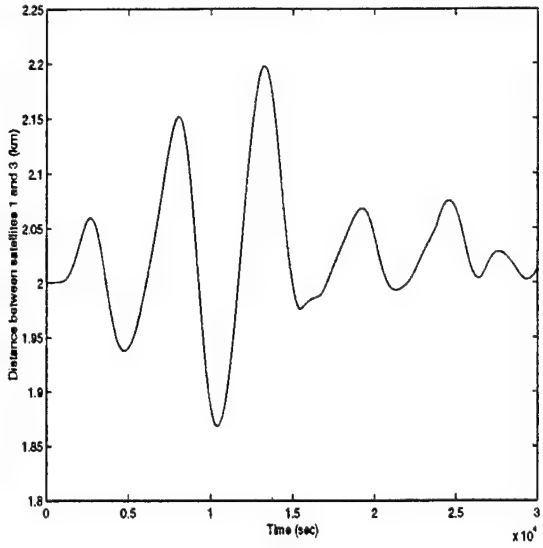


Figure 5.3: Distance between satellites 1 and 3: unrestricted burns, correction every other orbit

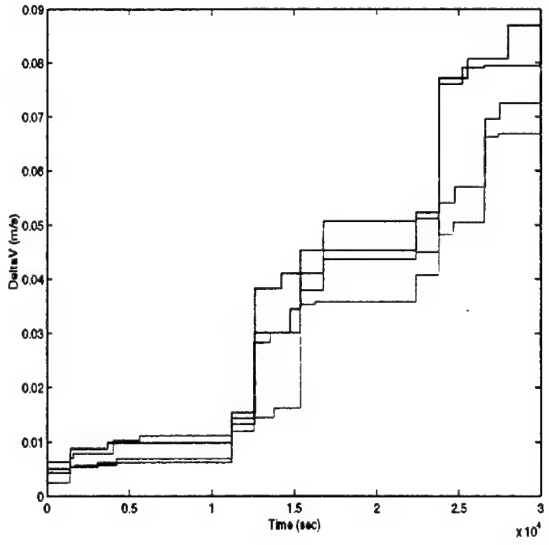


Figure 5.4: DeltaV: unrestricted burns, correction every other orbit

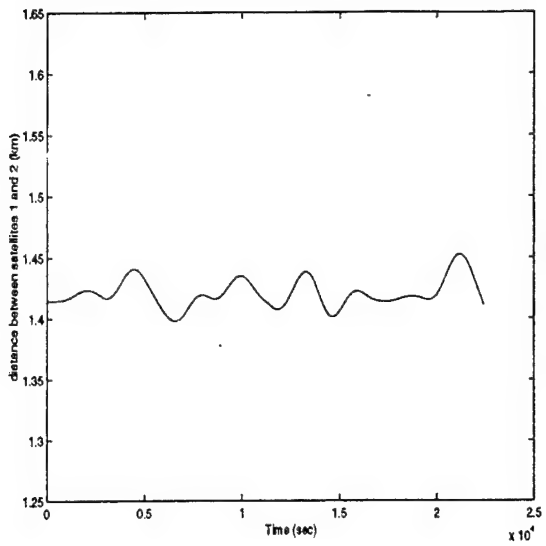


Figure 5.5: Distance between satellites 1 and 2: unrestricted burns, correction every orbit

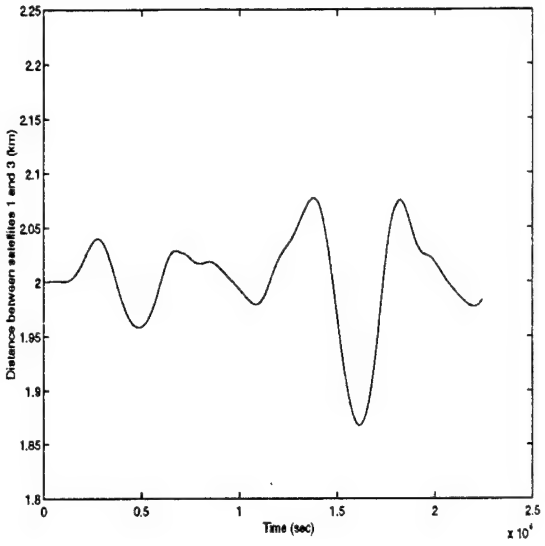


Figure 5.6: Distance between satellites 1 and 3: unrestricted burns, correction every orbit

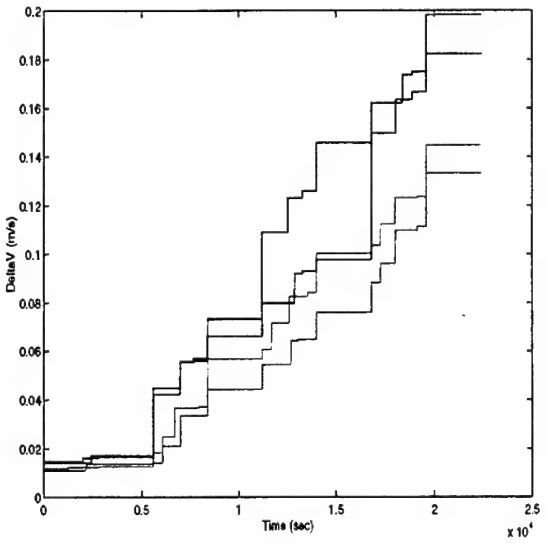


Figure 5.7: DeltaV: unrestricted burns, correction every orbit

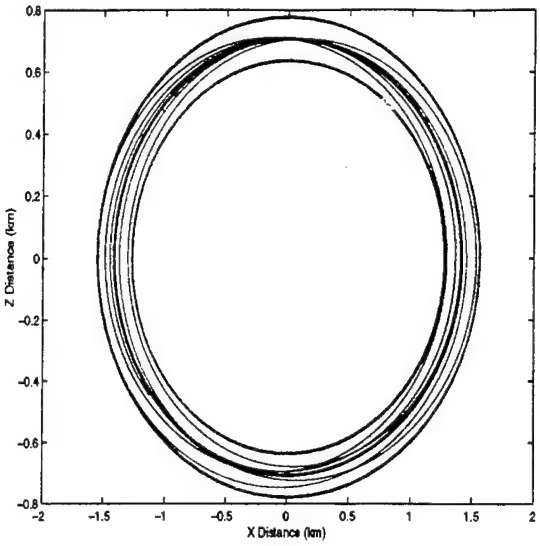


Figure 5.8: In-plane: unrestricted burns, correction every other orbit

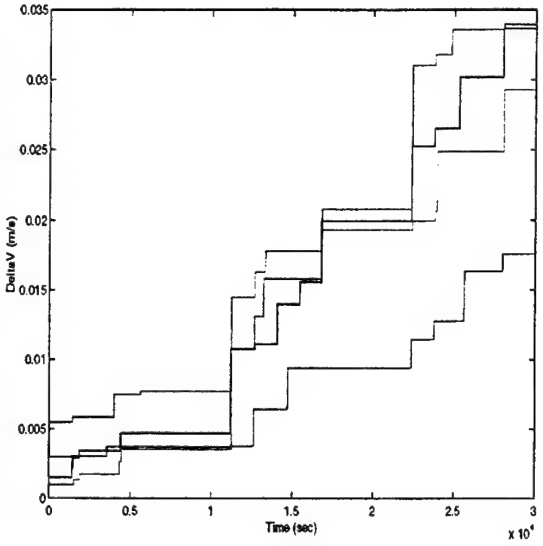


Figure 5.9: DeltaV: unrestricted burns, correction every other orbit

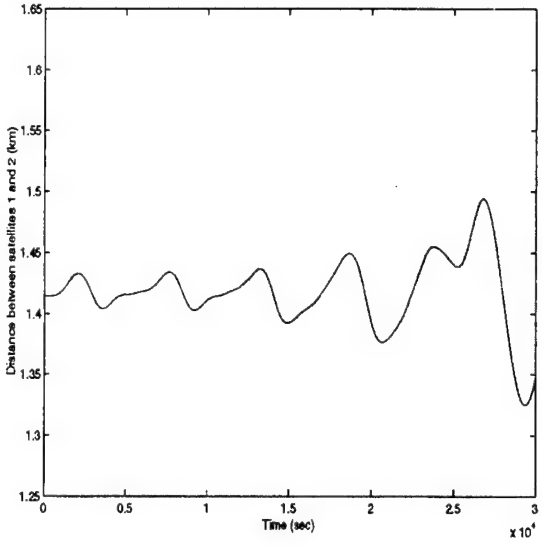


Figure 5.10: Distance between satellites 1 and 2: 10 minute burns, correction every other orbit

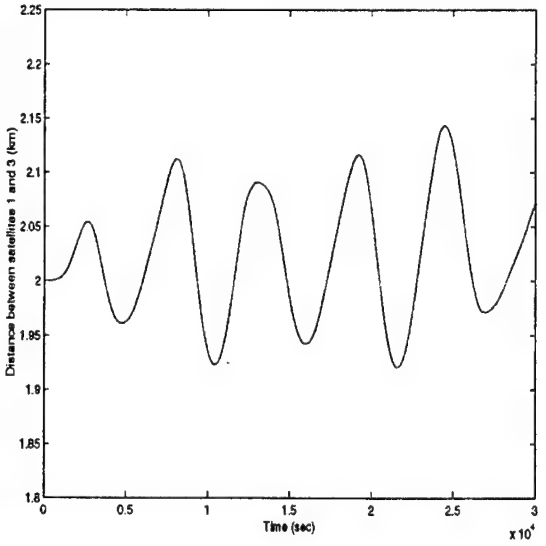


Figure 5.11: Distance between satellites 1 and 3: 10 minute burns, correction every other orbit

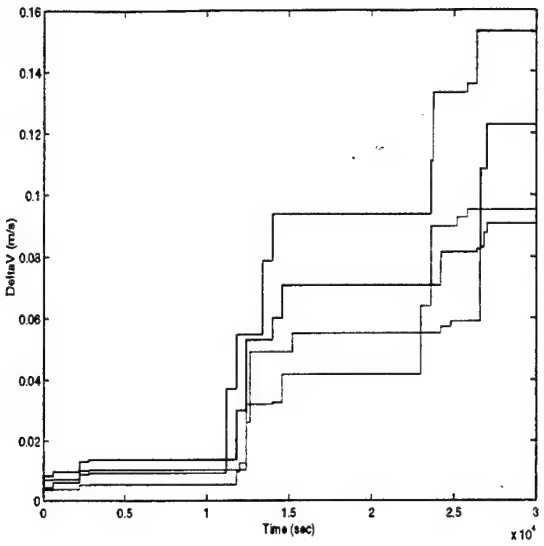


Figure 5.12: DeltaV: 10 minute burns, correction every other orbit

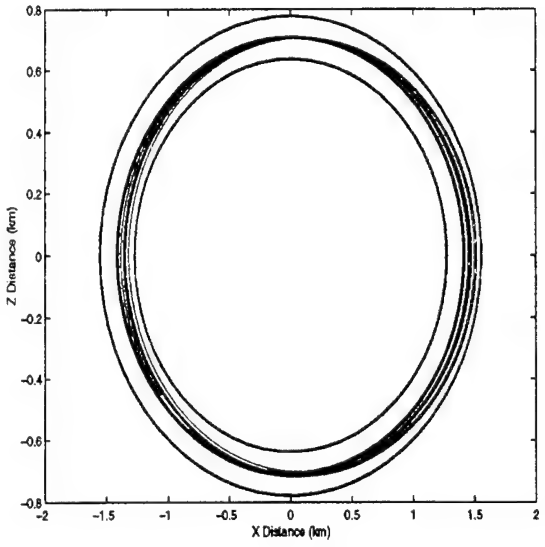


Figure 5.13: In-plane: 10 minute burns, correction every other orbit

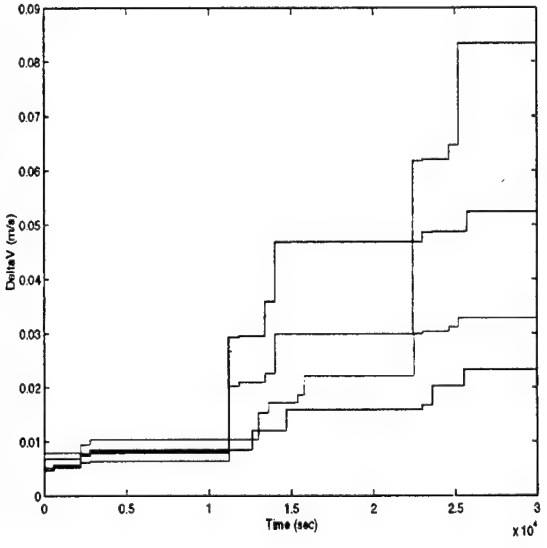


Figure 5.14: DeltaV: 10 minute burns, correction every other orbit

## 5.3 Effects of Relative Velocity Sensing Error

Another important factor in fuel consumption is the error in relative velocity sensing. To quantify this effect, we ran simulations in which the planner had access to the exact orbital model in order to exclude effects of mismatched dynamics. The baseline relative velocity error was taken to be 6mm/s. This is an optimistic (though possible) value for the error, given the technologies proposed for the demonstrator mission.

Figures 5.15, 5.16, and 5.17 represent the fuel usage of all four satellites for cases with baseline noise, double baseline noise, and half baseline noise. The total fuel usage grows linearly with the measurement noise. For baseline noise, note also that the effect of dynamics mismatch in the trajectory planning algorithms (as shown in the previous section) is similar in magnitude to the effect of noise. Although the former can be reduced by using more sophisticated nominal trajectories, the latter is a fundamental limitation. The results are summarized in Table 5.1.

Table 5.1: Effect of velocity sensing error

| Relative Velocity Error (mm/s) | Fuel Usage per Orbit (mm/s) | Endurance (days) |
|---|---|---|
| 3 | 9.6 | 420 |
| 6 | 15.6 | 260 |
| 12 | 25.9 | 157 |

## 5.4 Summary of Constellation Endurance for the Demonstrator Mission

The purpose of this study was to analyze the feasibility of the demonstrator mission objectives given the operational constraints. Since, in the demonstrator mission for Techsat21 the large out-of-plane formations are only to be flown for a few days, the results presented in Table 5.2 are very encouraging. However, when these simulation experiments were run we did not have the final set of operation guidelines for the constellation (which are somewhat still evolving). In particular, we did not consider the effect of doing burns along-track and across-track sequentially. The effect of having to use the engine for short intervals was also likely underestimated. We did not, however, have enough information from the engine supplier to fully understand these usage restrictions.

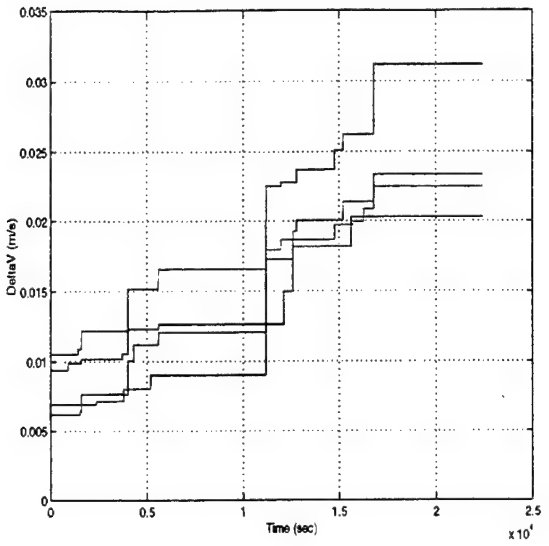


Figure 5.15: Fuel usage with relative velocity error of 6 mm/s

Table 5.2: Summary of performance with high-fidelity gravity model

| Formation | Burns | $\Delta$V per Orbit (mm/s) | Endurance (Days) |
|---|---|---|---|
| In-Plane | Unrestricted | 8.5 | 408 |
| | Two 10-min | 20.8 | 167 |
| Out-of-Plane | Unrestricted | 21.8 | 159 |
| | Two 10-min | 38.2 | 91 |

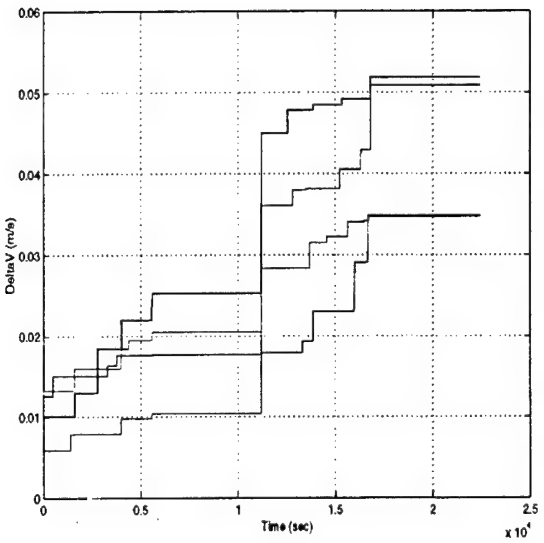


Figure 5.16: Fuel usage with relative velocity error of 12 mm/s

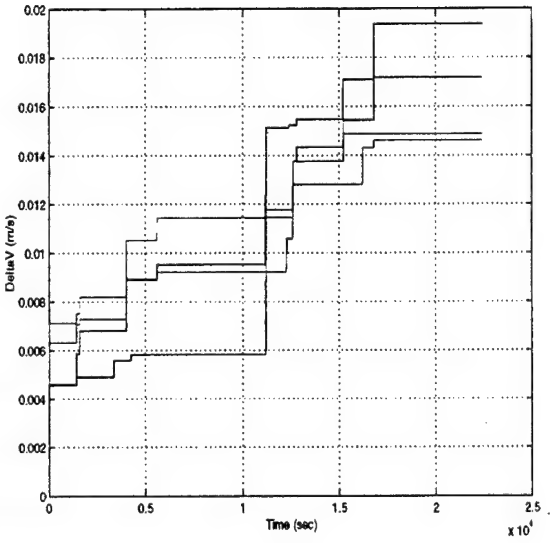


Figure 5.17: Fuel usage with relative velocity error of 3 mm/s

# Section 6 Trajectory Optimization

In this section we analyze how trajectories that meet the mission constraints can be generated. We follow an approach very similar to that presented in [4]. However, we used a different set of constraints that better match the objectives of the Techsat21 mission and our guidance laws.

## 6.1 Trajectory Parameterization

Io optimize the trajectory, it was divided into $n$ intervals. Within each time interval, each of the coordinates of the orbit was fit with a third-degree polynomial.

To compute initial values for the coefficients, we began with Euler orbits for all the satellites, satisfying the Hill conditions for periodic relative motion. Given the initial and final position and velocity for each interval, it is possible to determine the initial coefficients of the polynomial. Assume the initial and final positions of the satellite are represented by

$$R_o = c_o + c_1 t_1 + c_2 t_1^2 + c_3 t_1^3$$

$$R_f = c_o + c_1 t_2 + c_2 t_2^2 + c_3 t_2^3$$

Taking the derivative of tne position polynomials yields equations for the initial and final velocities of the satellites, which can be represented by

$$V_o = 0 + c_1 + 2c_2 t_1 + 3c_3 t_1^2$$

$$V_f = 0 + c_1 + 2c_2 t_2 + 3c_3 t_2^2$$

If $R_o, R_f, V_o, V_f$, and the initial and final times for the interval ($t_1$ and $t_2$) are known, it is possible to solve for the four coefficients of the polynomial by constructing a matrix such that

$$\begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 0 & 1 & 2t_1 & 3t_1^2 \\ 0 & 1 & 2t_2 & 3t_2^2 \end{bmatrix} \times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} R_o \\ R_f \\ V_o \\ V_f \end{bmatrix}$$

If this multiplication is repeated for each time interval in the orbit, the initial coefficients for the entire orbit can be calculated.

For the optimization routine to perform correctly, it was necessary to scale the parameters describing the trajectories so that they all had values in similar ranges and so that they all had similar effects on the cost function.

To achieve this we introduced a set of new variables denoted $d_0, d_1, d_2$, and $d_3$. We chose these variables such that

$$\frac{\partial x}{\partial d_0} \approx \frac{\partial x}{\partial d_1} \approx \frac{\partial x}{\partial d_2} \approx \frac{\partial x}{\partial d_3}$$

One way of achieving the correct scaling is through the following definitions (repeated for each coordinate, for each satellite, and for each time interval):

$$c_0 = c_{0i} + d_0$$

$$c_1 = c_{1i} + \frac{d_1}{t_2}$$

$$c_2 = c_{2i} + \frac{d_2}{t_2^2}$$

$$c_3 = c_{3i} + \frac{d_3}{t_2^3}$$

where $t_2$ is the length of each interval.

## 6.2 Constraints

We had two sets of constraints. One set imposed continuity of the orbits, while the other set imposed operational requirements on the resulting orbits for all satellites.

The first set of constraints required that the position and velocity of each satellite at the end of one interval be within a certain epsilon of the position and velocity of the satellite at the beginning of the following interval.

$$|R_1 - R_2|^2 \leq 0 \text{ and } |V_1 - V_2|^2 \leq 0$$

The other set of constraints assured that the constellation maintained a functional shape. For this experiment, we used a constellation of four satellites such that their projections on a horizontal plane centered at the reference lie on a fixed circle and form at every time a square. For a square formation to be "rigid," it is necessary to fix five of the six distances between the vertices. Thus,

the projection of the formation will remain in the same approximate shape by imposing that five of the six dimensions remain within a certain tolerance of their nominal values. For our experiments, we chose the five distances shown in Figure 6.1, and we allowed a 10% deviation from the nominal distance.

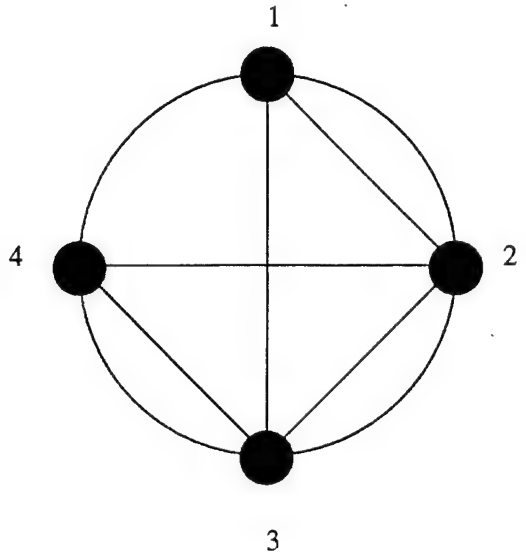


Figure 6.1: Constraints necessary to fix the cluster shape

For a 1km orbit, the constraint on each of the diagonal distances was

$$1.8 \leq d \leq 2.2$$

and the constraint corresponding to the edges was

$$\left(\sqrt{2} - .1\sqrt{2}\right) \leq d \leq \left(\sqrt{2} + .1\sqrt{2}\right)$$

To compute these distances, first we determined the center of the constellation by averaging the position of all four satellites. Then we projected all four satellites into the horizontal plane (normal to the radial direction) through the constellation center. For a half-orbit planning horizon, with 180 second intervals, there were 161 constraints ( n-1 constraints for each of the five shape constraints plus n-2 position and velocity constraints for each satellite).

## 6.3 Objective Function

Our objective was to minimize the amount of fuel needed to maintain a formation that meets the mission requirements. To do this, we used the "inversion" process described in [4] to compute the fuel necessary for a given formation. We measured fuel in units of $\Delta V$. To further simplify the process, we computed the total $\Delta V$ in one time interval as the acceleration due to the engines in the middle of the interval times. The accelerations were measured for a length of one time interval. For each of the satellites and for each of the coordinates, we had

$$U = \begin{bmatrix} 0 & 0 & 2 & 6t \end{bmatrix} \times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} - a_g$$

We compute the acceleration due to gravity ($a_g$) with the following formulas:

$$a_x = \ddot{x} = -\frac{\mu x}{r^3} + J_2 \frac{3}{2} r_e^2 \mu \left( \frac{5z^2 x}{r^7} - \frac{x}{r^5} \right)$$

$$a_y = \ddot{y} = -\frac{\mu y}{r^3} + J_2 \frac{3}{2} r_e^2 \mu \left( \frac{5z^2 y}{r^7} - \frac{y}{r^5} \right)$$

$$a_z = \ddot{z} = -\frac{\mu z}{r^3} - J_2 \frac{3}{2} r_e^3 \mu \left( \frac{3z}{r^6} - \frac{5z^3}{r^8} \right)$$

Finally, we computed our objective function as the sum of the squares of the $\Delta V$'s corresponding to all satellites, all directions, and all time intervals.

## 6.4 Optimization

The optimization routine used in this project was fmincon from the Matlab optimization package. Fmincon uses a sequential quadratic programming (SQP) algorithm. The number of variables depends on the time interval being used. We ran test cases for different planning horizons. For a planning horizon of a half orbit cut into 180 second intervals, we had 642 decision variables ( 4 satellites times 4 coefficients for each direction times 3 directions (x,y,z) times (n-1) where n is the number of time intervals.)

The results obtained for the half-orbit planning horizon are shown in Figures 6.2, 6.3, 6.4, and 6.5. Note that the maximum $\Delta V$ observed for one orbit is 0.9 mm/s. This corresponds to 4500 days of continuous operations in this 1km formation. Of course, this is the nominal $\Delta V$. The

actual fuel usage will be higher since it is significantly influenced by the effects discussed in the previous section. Note also that to maintain the Hill periodical orbits in the presence of J2 requires a $\Delta V$ of 40 mm/s per satellite.

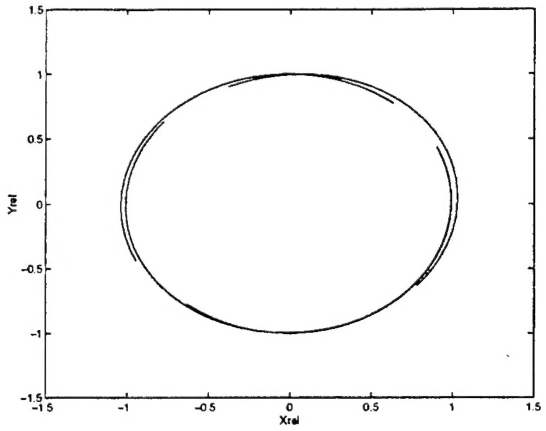


Figure 6.2: Fuel optimal trajectory shape

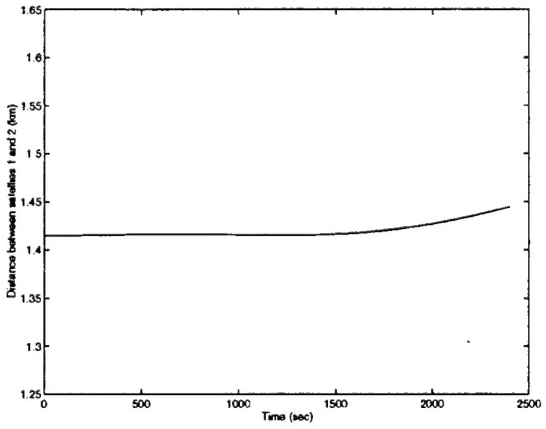


Figure 6.3: Fuel optimal trajectory distance between satellites 1 and 2

These results confirm the feasibility of the inversion approach to finding optimal trajectories presented in [4], even in the presence of the fairly large number of constraints required to impose all the operational requirements. The computation time for the problem run was several hours. (However, we computed the constraint gradients numerically, which significantly slowed down

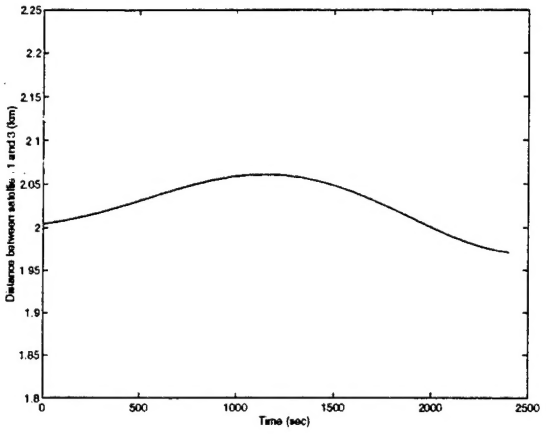


Figure 6.4: Fuel optimal trajectory distance between satellites 1 and 3

the computation.) Also, as for every optimization-based algorithm, guaranteeing convergence to a valid answer systematically is difficult.

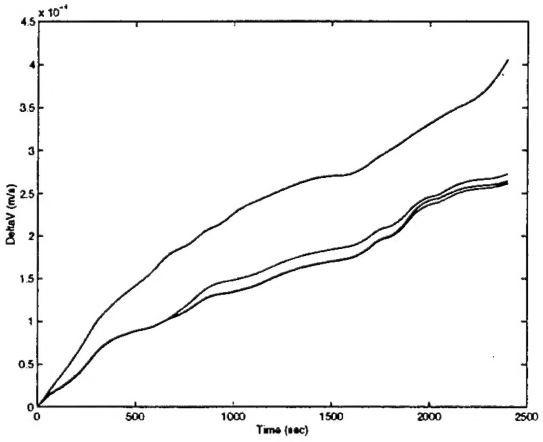


Figure 6.5: Fuel optimal trajectory $\Delta V$ for all satellites

# Bibliography


[1] Wei Kang and Ning Xi. Sensor-referenced motion planning and dynamic control for mobile robots. In Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics. Part 1 (of 5), pages 427–432, 1998.

[2] Leslie Lamport, Robert Shostak, and Marshall Peas. The byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4(3), 1982.

[3] J. Milnor. On the geometry of the Kepler problem. American Mathematical Monthly, pages 353–365, Jun/Jul 1983.

[4] Nicolas Petit, Mark B. Milam, and Richard M. Murray. Inversion based trajectory optimization. In Proceedings of the IFAC Symposium on Nonlinear Control Systems Design (NOLCOS), 2001.

[5] W. Walsh and M. Wellman. A market protocol for decentralized task allocation. In Third International Conference on Multiagent Systems, July 1998.

[6] M. Wellman. The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes. Computational Economics, 12(1):1–24, 1998.